

ReproBLAS: Reproducible BLAS

<http://bebop.cs.berkeley.edu/reproblas/>

James Demmel, **Nguyen Hong Diep**

SC'13 - Denver, CO

Nov 22, 2013

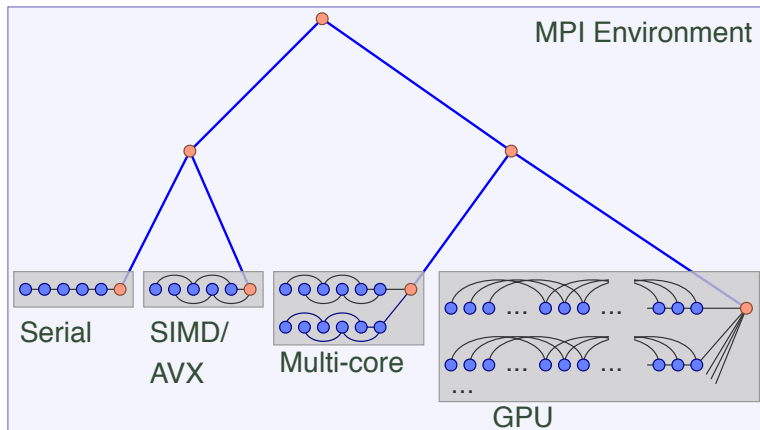
Reproducibility

Reproducibility: obtaining bit-wise identical results from different runs of the program on the same input data, regardless of different available resources.

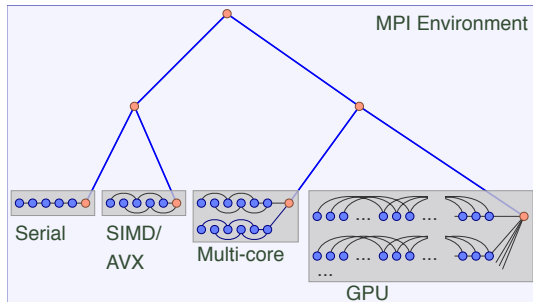
Cause of nonreproducibility: *not* by roundoff error but by the *non-determinism* of accumulative roundoff error.

Due to the *non-associativity* of floating point addition, accumulative roundoff errors depend on the order of evaluation, and therefore depend on available computing resources.

Sources of non-reproducibility

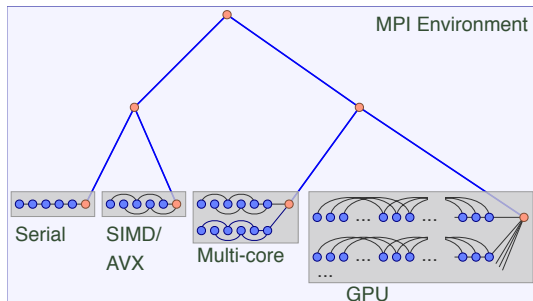


Sources of non-reproducibility



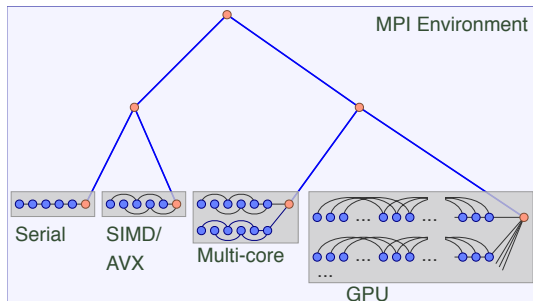
► number of MPI nodes

Sources of non-reproducibility



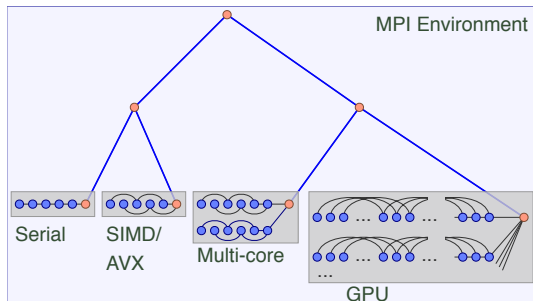
- ▶ number of MPI nodes
- ▶ MPI reduction tree shape

Sources of non-reproducibility



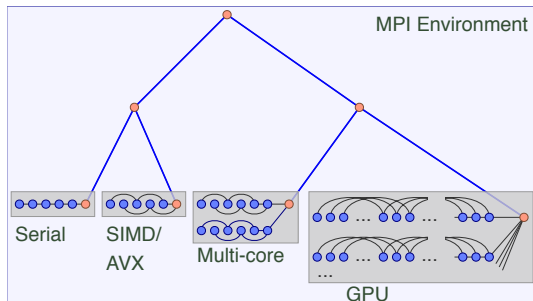
- ▶ number of MPI nodes
- ▶ MPI reduction tree shape
- ▶ number of cores per node

Sources of non-reproducibility



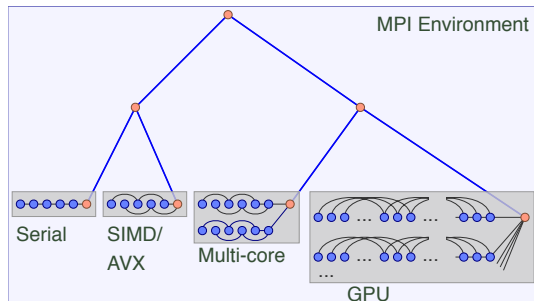
- ▶ number of MPI nodes
- ▶ MPI reduction tree shape
- ▶ number of cores per node
- ▶ data path (1,2,4,...)

Sources of non-reproducibility



- ▶ number of MPI nodes
- ▶ MPI reduction tree shape
- ▶ number of cores per node
- ▶ data path (1,2,4,...)
- ▶ data ordering

Sources of non-reproducibility

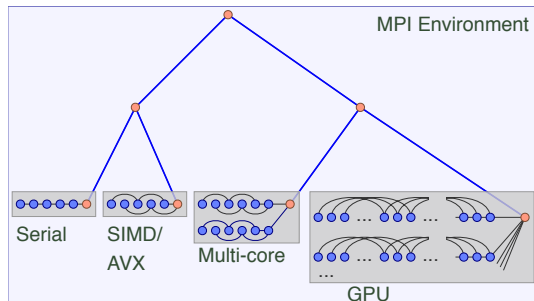


- ▶ number of MPI nodes
- ▶ MPI reduction tree shape
- ▶ number of cores per node
- ▶ data path (1,2,4,...)
- ▶ data ordering

Related work:

- ▶ Intel MKL 11.0 with BWR: reproducible with fixed number of processors, and fixed data alignment,

Sources of non-reproducibility



- ▶ number of MPI nodes
- ▶ MPI reduction tree shape
- ▶ number of cores per node
- ▶ data path (1,2,4,...)
- ▶ data ordering

Related work:

- ▶ Intel MKL 11.0 with BWR: reproducible with fixed number of processors, and fixed data alignment,
- ▶ Hardware (NIC) for reproducible reduction operator.

Proposed solution

Goal: obtaining *deterministic* roundoff errors.

Exact arithmetic: unacceptably expensive in both long-word arithmetics and communication

Fixed point arithmetic:

- ▶ limited in exponent range
- ▶ limited in value range

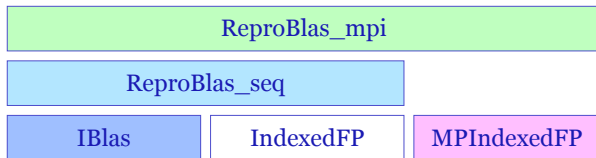
Indexed floating point: ¹

- ▶ Use floating point numbers to represent extended-precision fixed point,
- ▶ Exponent is of format $i \cdot W$ and is adjusted (on-line) according to the maximum absolute value of input,
- ▶ Can add up to 2^{60} floating point numbers,
- ▶ No extra reduction required for distributed environment,

¹J. Demmel and H. D. Nguyen, *Fast Reproducible Floating-Point Summation*, ARITH21, Austin, Texas, April 7–10, 2013

ReproBLAS: Reproducible Basic Linear Algebra Subprograms

URL: <http://bebop.cs.berkeley.edu/reproblas>



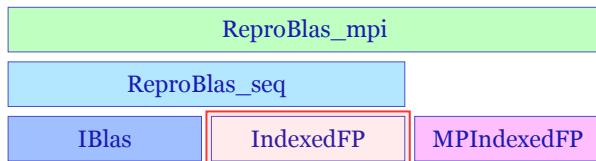
Features:

- ▶ Reproducibility,
- ▶ Accuracy: no severe accuracy loss,
- ▶ Reasonable arithmetic cost,
- ▶ Minimize communication: only one reduction per sum.

Requirements: **ROUND-TO-NEAREST** and *no overflow*

ReproBLAS: Reproducible Basic Linear Algebra Subprograms

URL: <http://bebop.cs.berkeley.edu/reproblas>

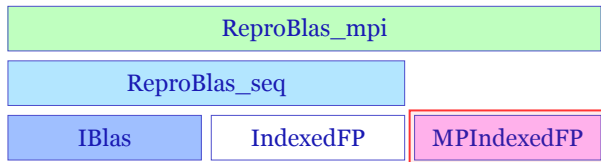


IndexedFP implement arithmetics for Indexed Floating Point:

- ▶ Data types: Idouble, Ifloat, dIcomplex, sIcomplex
- ▶ Conversion to: dconv2I, sconv2I, zconv2I, cconv2I
- ▶ Conversion from: Iconv2d, Iconv2f, Iconv2z, Iconv2c
- ▶ Addition: dIAdd, sIAdd, zIAdd, cIAdd
- ▶ Cross-type operations: dIAddd, sIAddf, zIAddz, cIAddc

ReproBLAS: Reproducible Basic Linear Algebra Subprograms

URL: <http://bebop.cs.berkeley.edu/reproblas>

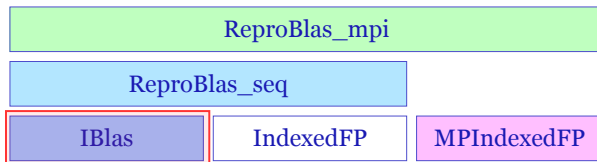


MPIndexedFP is an MPI wrapper for IndexedFP:

- ▶ Data types: `MPI_IDOUBLE`, `MPI_IFLOAT`, `MPI_IDOUBLE_COMPLEX`, `MPI_ICOMPLEX`
- ▶ Reduction operators: `MPI_RSUM`, `MPI_RNRM2`

ReproBLAS: Reproducible Basic Linear Algebra Subprograms

URL: <http://bebop.cs.berkeley.edu/reproblas>

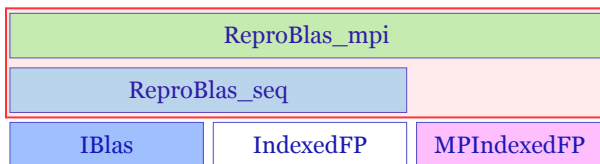


IBlas (Indexed Blas) provides performance-optimized kernel routines which take in vectors in native data type and return result in Indexed Floating Point format

- ▶ $\{s|d|c|z\}asumI$
- ▶ $\{s|d|c|z\}sumI$
- ▶ $\{s|d|c|z\}nrm2I$
- ▶ $\{s|d|c|z\}dot\{c|u\}I$

ReproBLAS: Reproducible Basic Linear Algebra Subprograms

URL: <http://bebop.cs.berkeley.edu/reproblas>



Current version only supports level 1 routines for 4 basic data types:

- ▶ $\{r|pr\}\{s|d|c|z\}asum$
- ▶ $\{r|pr\}\{s|d|c|z\}sum$
- ▶ $\{r|pr\}\{s|d|c|z\}nrm2$
- ▶ $\{r|pr\}\{s|d|c|z\}dot \{c|u\}$

Development status

x: complete o: in progress

	Single-threaded			Multi-threaded		GPU	MPI	...
	Scalar	SSE	AVX	OMP	pthread			
BLAS 1	asum	x	x	o	o		x	
	sum	x	x	o	o		x	
	dot	x	x	o	o		x	
	nrm2	x	x	o	o		x	
2	gemv	o	o					
	trsv	o	o					
3	gemm							
	trsm							

Future work: LAPACK, hardware support, non-linear operations, ...